Other Useful Packages

The following packages are contributions from users. The module descriptions are shown in the CDAT Utilities Reference Guide cdat_utilities.pdf. They are provided "as−is" and may not be supported unless the package is considered useful by a large user community.

Interface to Spherepack

This package contains a Python interface to the subroutine library Spherepack. To see list of functions type

```
% pydoc -w sphere
```

Interface to Regridpack

This package contains a Python interface to the subroutine library regridpack. For further details type:

```
% pydoc -w adamsregrid
```

Empirical Orthogonal Functions

Available in the eof package. Calculates Empirical Orthogonal Functions of either one variable or two variables jointly. For more documentation type:

```
% pydoc -w eof
```

Interface to the L−moments library

An interface to an L−moments library by J. R. M. Hosking. To see list of functions type:

```
% pydoc -w lmoments
```

Interface to the ngmath library

The ngmath library is a collection of interpolators and approximatorsfor one−dimensional, two−dimensional and three−dimensional data. The packages, which were obtained from NCAR, are:

- **natgrid** − a two−dimensional random data interpolation package based on Dave Watson's nngridr. NOT built by default in CDAT due to compile problems on some platforms. Works on linux.
- **dsgrid** − a three−dimensional random data interpolator based on a simple inverse distance weighting algorithm.
- **fitgrid** − an interpolation package for one−dimensional and two−dimensional gridded data based on Alan Cline's Fitpack. Fitpack uses splines under tension to interpolate in one and two dimensions. NOT IN CDAT.
- **csagrid** − an approximation package for one−dimensional, two−dimensional and three−dimensional random data based on David Fulker's Splpack. csagrid uses cubic splines to calculate its approximation function.

Using existing Fortran code

Pyfort

Pyfort is a tool for connecting Fortran (Fortran90) routines to Python (www.python.org). Pyfort translates an input file that describes the Fortran functions and subroutines you wish to access from Python into a C language source file defining a Python module. Fortran was changed significantly by the introduction of the Fortran 90 standard. We will use the phrase "modern Fortran" to indicate versions of Fortran from Fortran 90 onwards. Pyfort's input uses a syntax that is a subset of the modern Fortran syntax for declaring routines and their arguments. The current release does not yet support modern Fortran's "explicit−interface" routines. However, the tool was designed with this in mind for a future release. Pyfort can in most cases also build and install the extension you create.

The Pyfort project page at SourceForge contains documentation and releases. It is: http://sourceforge.net/projects/pyfortran

F2PY (previously known as fpig)

Writing Python C/API wrappers for Fortran routines can be a very tedious task, especially if a Fortran routine takes more than 20 arguments but only few of them are relevant for the problems that they solve. Pearu Petersen has developed a tool that generates the C/API modules containing wrapper functions of Fortran routines. This tool is called F2PY Fortran to Python Interface Generator. It is completely written in Python language and can be called from the command line as f2py. F2PY is released under the terms of GNU LGPL. The F2PY package and documentation can be downloaded from http://cens.ioc.ee/projects/f2py2e/

Migrating from GrADS (grads)

The grads module supplies an interface to CDMS that will be familiar to users of GrADS. See the CDAT website for documentation.

ort

Read data from an Oort file.

trends

Computes variance estimate taking auto−correlation into account.

pyclimate

This package − also python based, is written and maintained at the Universidad del PaÃ-s Vasco in Bilbao, Spain. It provides useful statistical functions for climate applications.